In [49]:
```
import pandas as pd
pd.__version__
```

Out[49]:
```
'2.2.2'
```

In [51]:
```
emp=pd.read_excel(r'D:\datascience&AI notes\Rawdata.xlsx') #loading raw data fil
```

In [53]:
```
emp
```

Out[53]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience#$	34 years	Mumbai	5^00#0	2+
1	Teddy^	Testing	45' yr	Bangalore	10%%000	<3
2	Uma#r	Dataanalyst^^#	NaN	NaN	1$5%000	4>
3	Jane	Ana^^lytics	NaN	Hyderbad	2000^0	NaN
4	Uttam*	Statistics	67-yr	NaN	300^00	5+
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [55]:
```
id(emp)
```

Out[55]:
```
1997803830448
```

In [57]:
```
emp.columns
```

Out[57]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [59]:
```
emp.shape
```

Out[59]:
```
(6, 6)
```

In [61]:
```
emp.head
```

Out[61]:
```
<bound method NDFrame.head of      Name          Domain       Age   Location
Salary   Exp
0    Mike   Datascience#$  34 years     Mumbai   5^00#0   2+
1  Teddy^         Testing    45' yr  Bangalore  10%%000   <3
2   Uma#r  Dataanalyst^^#       NaN        NaN  1$5%000  4>
3    Jane     Ana^^lytics       NaN   Hyderbad   2000^0  NaN
4  Uttam*      Statistics     67-yr        NaN   300^00  5+
5     Kim             NLP      55yr      Delhi  6000^$0  10+>
```

In [63]:
```
emp.tail
```

Out[63]:
```
<bound method NDFrame.tail of      Name          Domain       Age   Location
Salary   Exp
0    Mike   Datascience#$  34 years     Mumbai   5^00#0   2+
1  Teddy^         Testing    45' yr  Bangalore  10%%000   <3
2   Uma#r  Dataanalyst^^#       NaN        NaN  1$5%000  4>
3    Jane     Ana^^lytics       NaN   Hyderbad   2000^0  NaN
4  Uttam*      Statistics     67-yr        NaN   300^00  5+
5     Kim             NLP      55yr      Delhi  6000^$0  10+>
```

In [65]:

```
emp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      6 non-null      object
 1   Domain    6 non-null      object
 2   Age       4 non-null      object
 3   Location  4 non-null      object
 4   Salary    6 non-null      object
 5   Exp       5 non-null      object
dtypes: object(6)
memory usage: 420.0+ bytes
```

In [67]:

```
emp.isnull() # if data miss returns true else false
```

Out[67]:

	Name	Domain	Age	Location	Salary	Exp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	True	True	False	False
3	False	False	True	False	False	True
4	False	False	False	True	False	False
5	False	False	False	False	False	False

In [69]:

```
emp.isna() #isnull &isna both are same
```

Out[69]:

	Name	Domain	Age	Location	Salary	Exp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	True	True	False	False
3	False	False	True	False	False	True
4	False	False	False	True	False	False
5	False	False	False	False	False	False

In [71]:

```
emp.isnull().sum()
```

Out[71]:

```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [73]:

```
emp.columns
```

Out[73]:

```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [75]:

```
emp
```

Out[75]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience#$	34 years	Mumbai	5^00#0	2+
1	Teddy^	Testing	45' yr	Bangalore	10%%000	<3
2	Uma#r	Dataanalyst^^#	NaN	NaN	1$5%000	4>
3	Jane	Ana^^lytics	NaN	Hyderbad	2000^0	NaN
4	Uttam*	Statistics	67-yr	NaN	300^00	5+
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [77]:

```
emp['Name']
```

Out[77]:

```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [192…

```
emp['Name']=emp['Name'].str.replace(r'\W','',regex=True)
```

In [194…

```
emp['Name']
```

Out[194…

```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [196…

```
emp['Domain']=emp['Domain'].str.replace(r'\W','',regex=True)
```

In [198…

```
emp['Domain']
```

Out[198…

```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [200…

```
emp['Age']=emp['Age'].str.replace(r'\W','',regex=True)
```

In [202…

```
emp['Age']
```

Out[202…

```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [204…

```
emp['Location']=emp['Location'].str.replace(r'\W','',regex=True)
```

In [206…

```
emp['Location']
```

Out[206…

```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [208…

```
emp['Salary']=emp['Salary'].str.replace(r'\W','',regex=True)
```

In [210…

```
emp['Salary']
```

Out[210…

```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [212…

```
emp['Exp']=emp['Exp'].str.replace(r'\W','',regex=True)
```

In [214…

```
emp['Exp']
```

Out[214…

```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [111…

```
emp['Age']=emp['Age'].str.extract('(\d+)')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\Dell\AppData\Local\Temp\ipykernel_13944\3771958390.py:1: SyntaxWarning:
invalid escape sequence '\d'
  emp['Age']=emp['Age'].str.extract('(\d+)')
```

In [216…

```
emp['Age']
```

Out[216…

```
0      34
1      45
2     NaN
3     NaN
4      67
5      55
Name: Age, dtype: object
```

In [115…

```
emp #cleaned all data set using str replace,extract
```

Out[115…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Uma r	Dataanalyst	NaN	NaN	15000	4
3	Jane	Ana lytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [117…

```
clean_data=emp.copy()
```

In [119…

```
clean_data
```

Out[119…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Uma r	Dataanalyst	NaN	NaN	15000	4
3	Jane	Ana lytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [121…

```
clean_data.isnull().sum()
```

Out[121…

```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [123…

```
clean_data['Age']
```

Out[123…

```
0      34
1      45
2     NaN
3     NaN
4      67
5      55
Name: Age, dtype: object
```

In [125…

```
import numpy as np
```

In [127…

```
clean_data['Age']=clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age
```

In [129…

```
clean_data['Age']
```

Out[129…

```
0        34
1        45
2     50.25
3     50.25
4        67
5        55
Name: Age, dtype: object
```

In [131…

```
clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp
```

In [133…

```
clean_data['Location'].isnull().sum()
```

Out[133…

```
2
```

In [135…

```
clean_data['Location']=clean_data['Location'].fillna(clean_data['Location'].mode
```

In [137…

```
clean_data['Location']
```

Out[137…

```
0        Mumbai
1     Bangalore
2     Bangalore
3      Hyderbad
4     Bangalore
5         Delhi
Name: Location, dtype: object
```

In [139…

```
clean_data
```

Out[139…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Uma r	Dataanalyst	50.25	Bangalore	15000	4
3	Jane	Ana lytics	50.25	Hyderbad	20000	4.8
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [141…

```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      6 non-null      object
 1   Domain    6 non-null      object
 2   Age       6 non-null      object
 3   Location  6 non-null      object
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
dtypes: object(6)
memory usage: 420.0+ bytes
```

In [143…

```
clean_data['Age']=clean_data['Age'].astype(int)#converting variable in to numeri
```

In [145…

```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      6 non-null      object
 1   Domain    6 non-null      object
 2   Age       6 non-null      int32 
 3   Location  6 non-null      object
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
dtypes: int32(1), object(5)
memory usage: 396.0+ bytes
```

In [147…

```
clean_data['Exp']=clean_data['Exp'].astype(int)
clean_data['Salary']=clean_data['Salary'].astype(int)
```

In [149…

```
clean_data['Name']=clean_data['Name'].astype('category')
clean_data['Domain']=clean_data['Domain'].astype('category')
clean_data['Location']=clean_data['Location'].astype('category')
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype   
---  ------    --------------  -----   
 0   Name      6 non-null      category
 1   Domain    6 non-null      category
 2   Age       6 non-null      int32   
 3   Location  6 non-null      category
 4   Salary    6 non-null      int32   
 5   Exp       6 non-null      int32   
dtypes: category(3), int32(3)
memory usage: 866.0 bytes
```

In [151…

```
clean_data.to_csv('clean_data.csv') # coverting clean data excel to csv file
```

In [153…

```
import os
os.getcwd() #creating clean data file in c directory
```

Out[153…

```
'C:\\Users\\Dell'
```

In [155…

```
import matplotlib.pyplot as plt # visualization
import seaborn as sns
```

In [157…

```
import warnings
warnings.filterwarnings('ignore')
```

In [159…

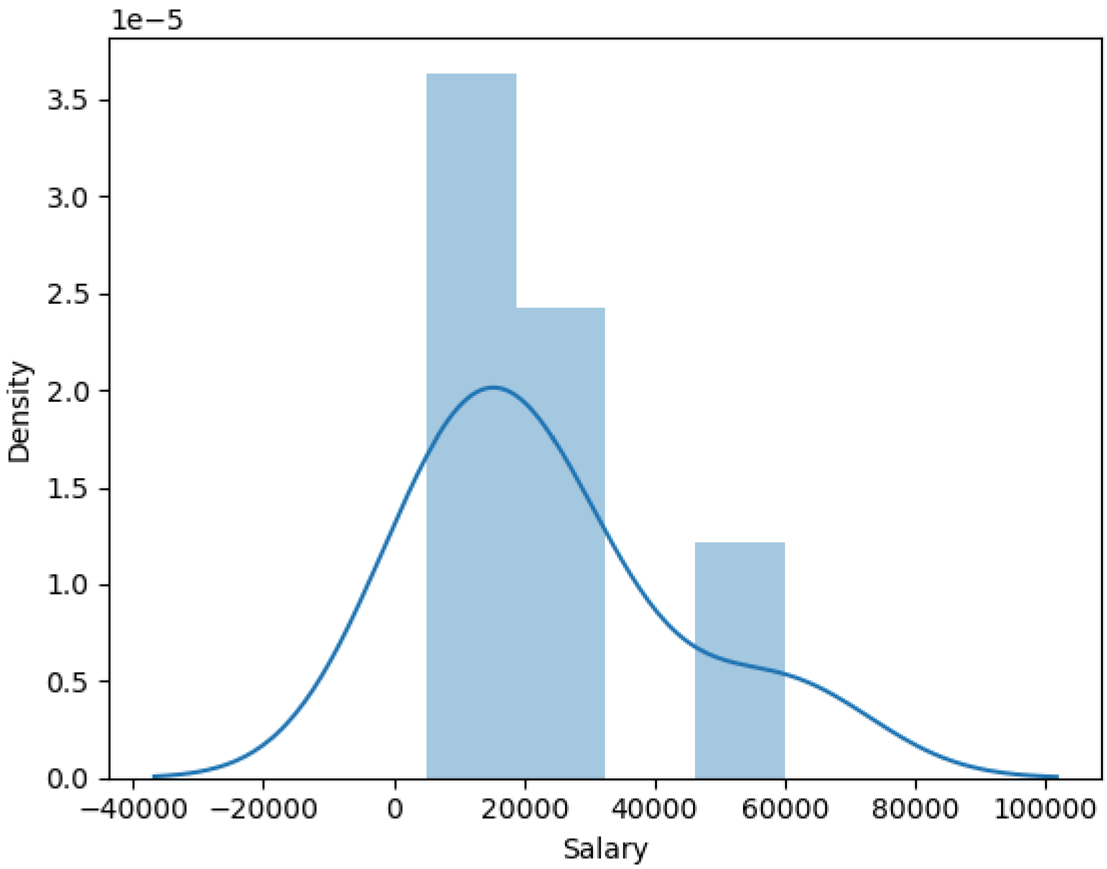
```
clean_data['Salary']
```

Out[159…

```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

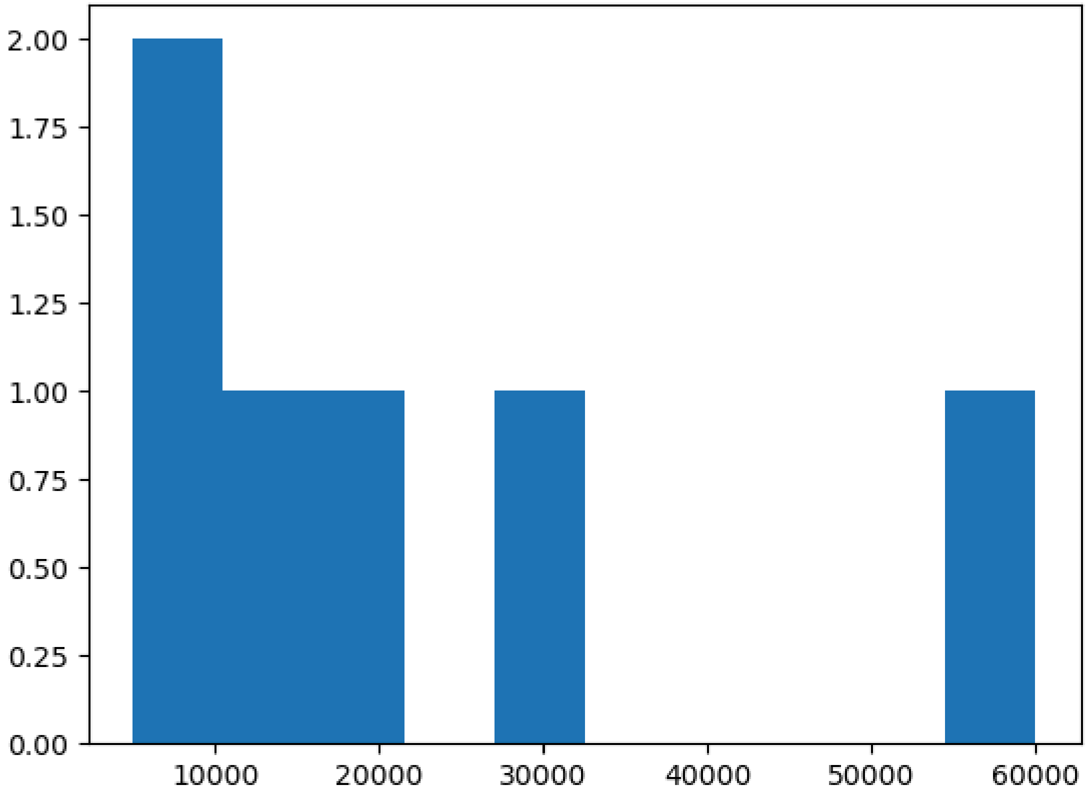
In [161…

```
vis1 = sns.distplot(clean_data['Salary'])#uni vaient ploting
```



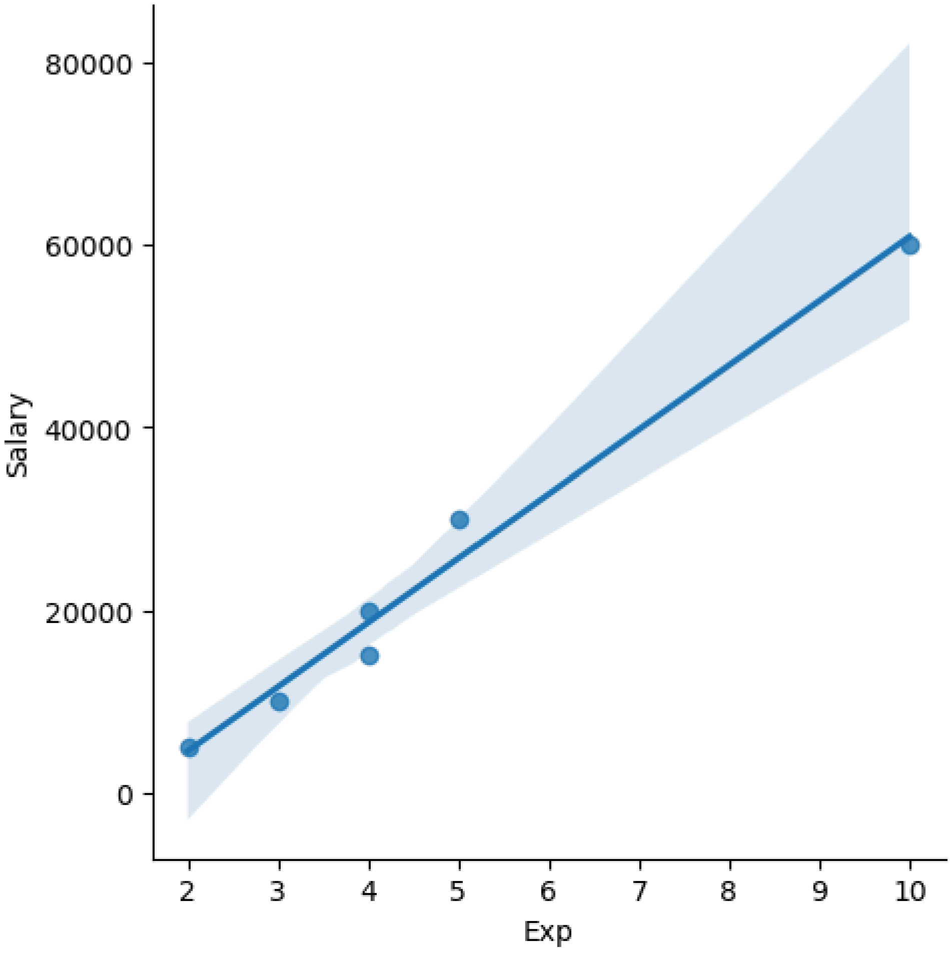
In [164…

```
vis2 = plt.hist(clean_data['Salary'])#outlier identifier
```



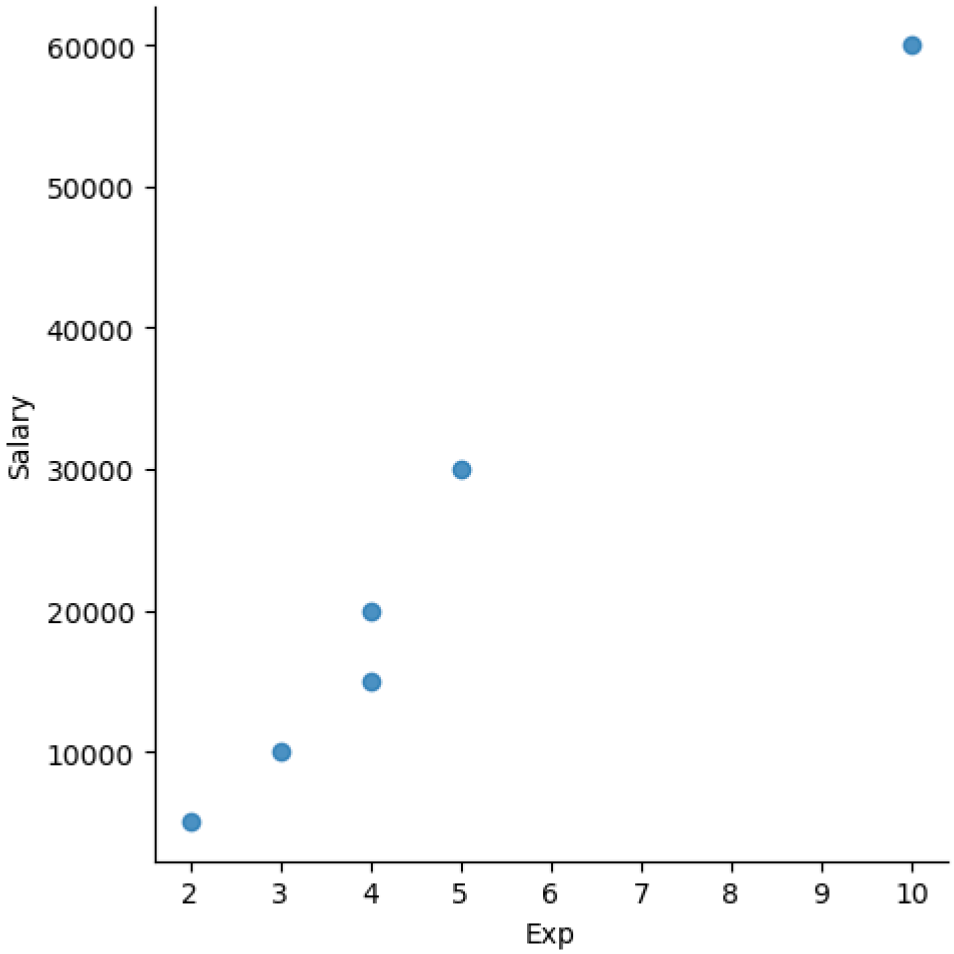
In [166…

```
vis4 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary') #bivariet ploting
```



In [168…

```
vis5 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary', fit_reg = False)
```



In [170…

```
clean_data.columns
```

Out[170…

```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [172…

```
X_iv = clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']] #idetifying inde
```

In [174…

```
X_iv
```

Out[174…

	Name	Domain	Age	Location	Exp
0	Mike	Datascience	34	Mumbai	2
1	Teddy	Testing	45	Bangalore	3
2	Uma r	Dataanalyst	50	Bangalore	4
3	Jane	Ana lytics	50	Hyderbad	4
4	Uttam	Statistics	67	Bangalore	5
5	Kim	NLP	55	Delhi	10

In [176…

```
y_dv = clean_data[['Salary']] #identifying dependent variable
y_dv
```

Out[176…

	Salary
0	5000
1	10000
2	15000
3	20000
4	30000
5	60000

In [178…

```
clean_data
```

Out[178…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Uma r	Dataanalyst	50	Bangalore	15000	4
3	Jane	Ana lytics	50	Hyderbad	20000	4
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [180…

```
imputation = pd.get_dummies(clean_data) # creatin variables using labeling ,dumm
```

In [184…

```
imputation
```

Out[184…

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Uma r	N
0	34	5000	2	False	False	True	False	False	
1	45	10000	3	False	False	False	True	False	
2	50	15000	4	False	False	False	False	True	
3	50	20000	4	True	False	False	False	False	
4	67	30000	5	False	False	False	False	False	
5	55	60000	10	False	True	False	False	False	

In [188…

```
clean_data.columns
```

Out[188…

```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [190…

```
imputation.columns
```

```
Out[190…  Index(['Age', 'Salary', 'Exp', 'Name_Jane', 'Name_Kim', 'Name_Mike',
                 'Name_Teddy ', 'Name_Uma r', 'Name_Uttam ', 'Domain_Ana  lytics',
                 'Domain_Dataanalyst   ', 'Domain_Datascience  ', 'Domain_NLP',
                 'Domain_Statistics', 'Domain_Testing', 'Location_Bangalore',
                 'Location_Delhi', 'Location_Hyderbad', 'Location_Mumbai'],
                dtype='object')
```

```
In [ ]:
```